TCP as a Reliable Transport

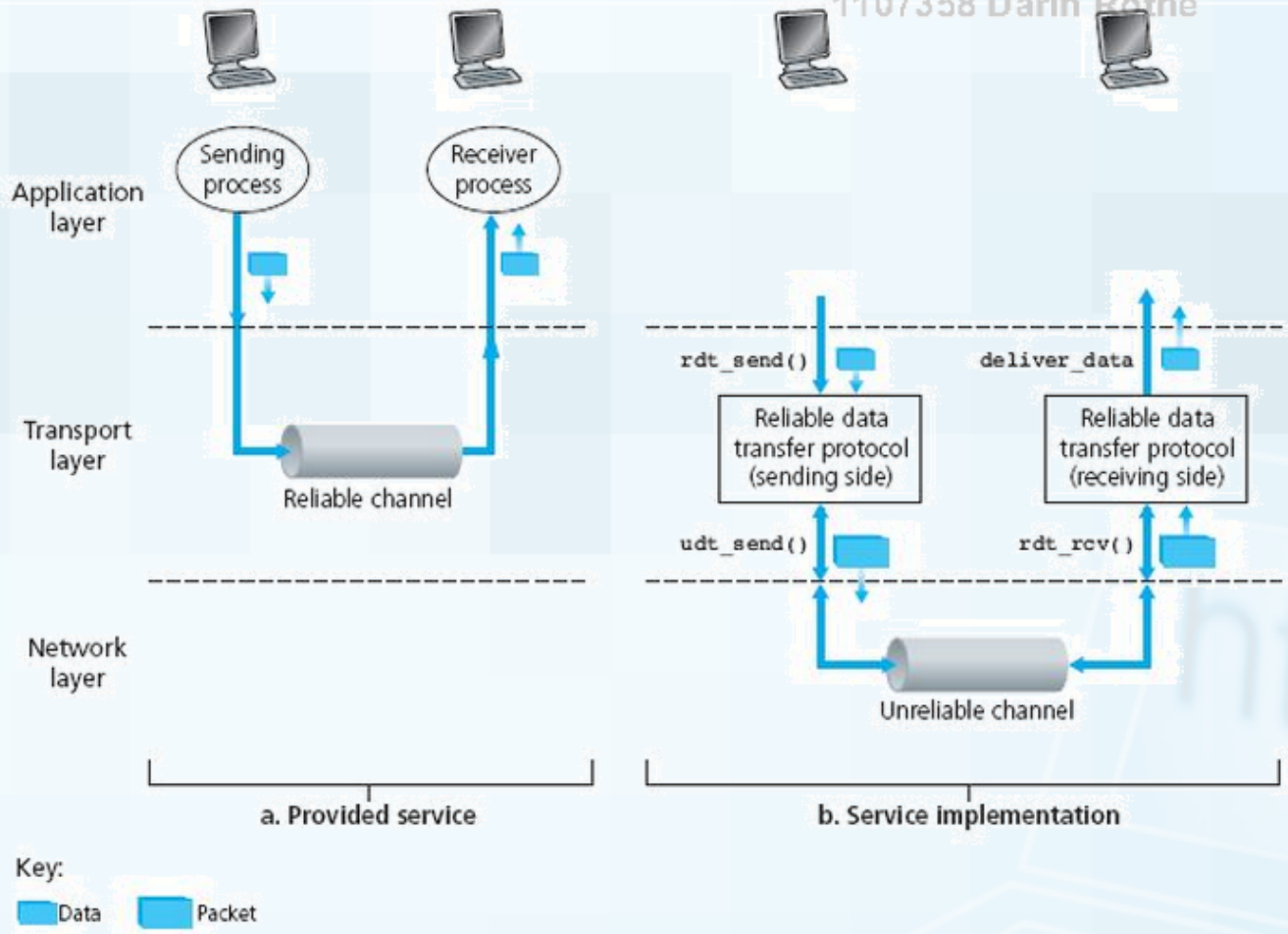


Figure 3.8 ♦ Reliable data transfer: Service model and service implementation

How things can go wrong…

- Lost packets
- Corrupted packets
- Reordered packets
- …Malicious packets…

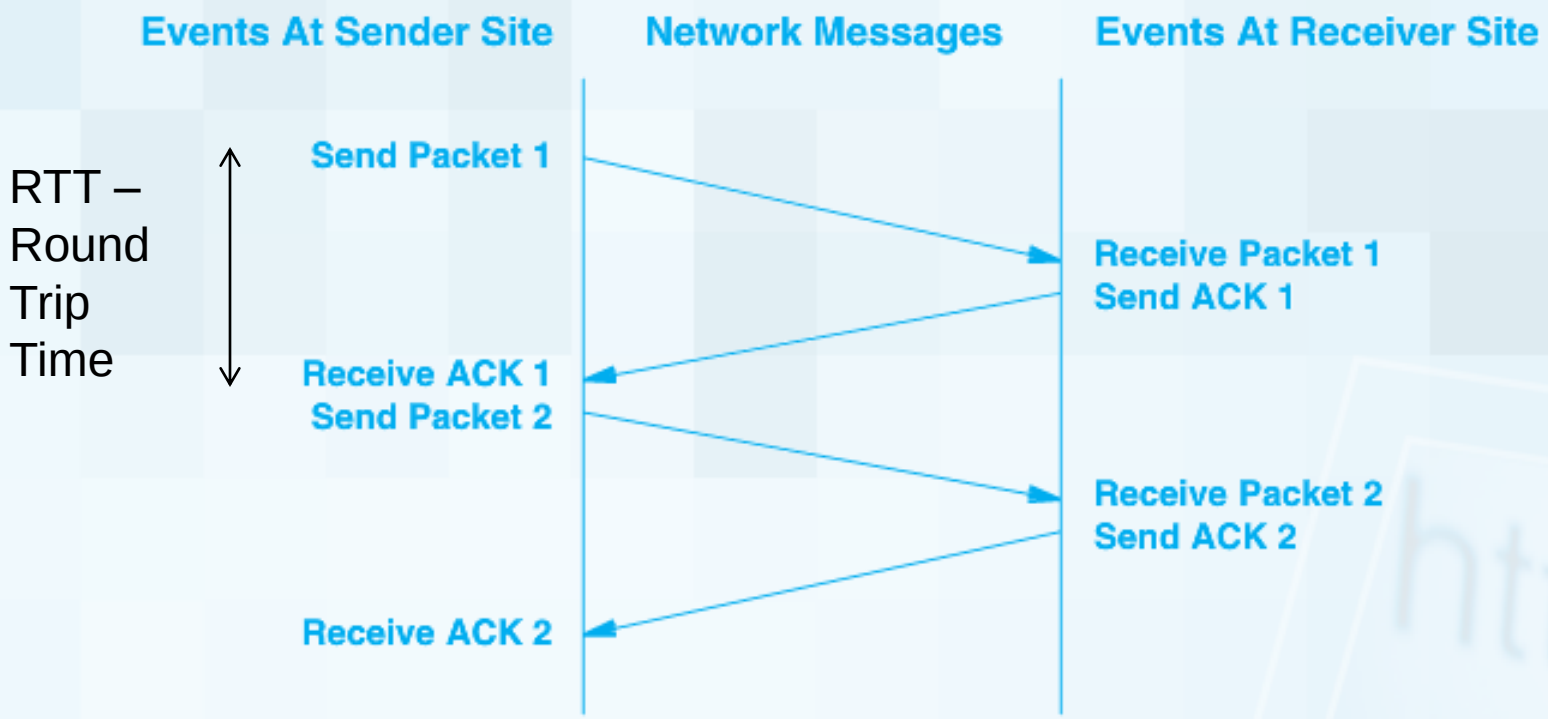
Requirements for Reliability

- Error Detection
- Receiver Feedback
- Retransmission

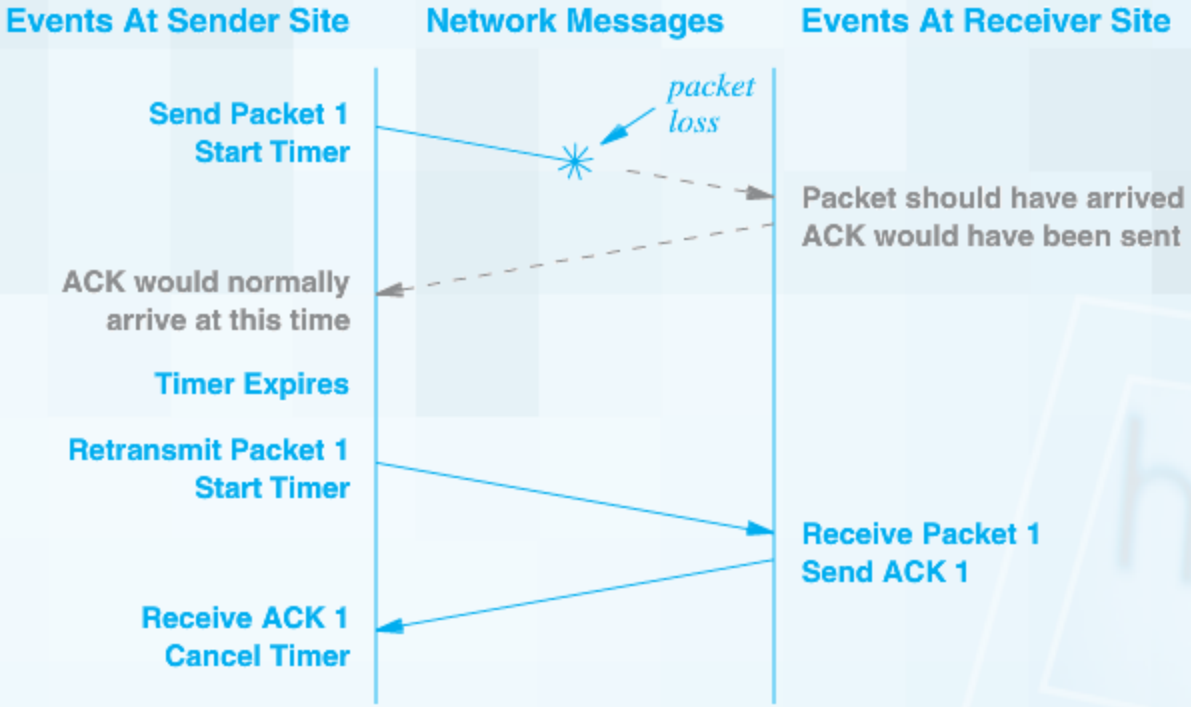
Requirements for Reliability

- Error Detection
  - Checksum
- Receiver Feedback
  - ACK – acknowledgment
  - NAK – negative acknowledgment
    - Also missing ACK
- Retransmission
  - Sender resends segment with NAK or missing ACK

Stop and Wait

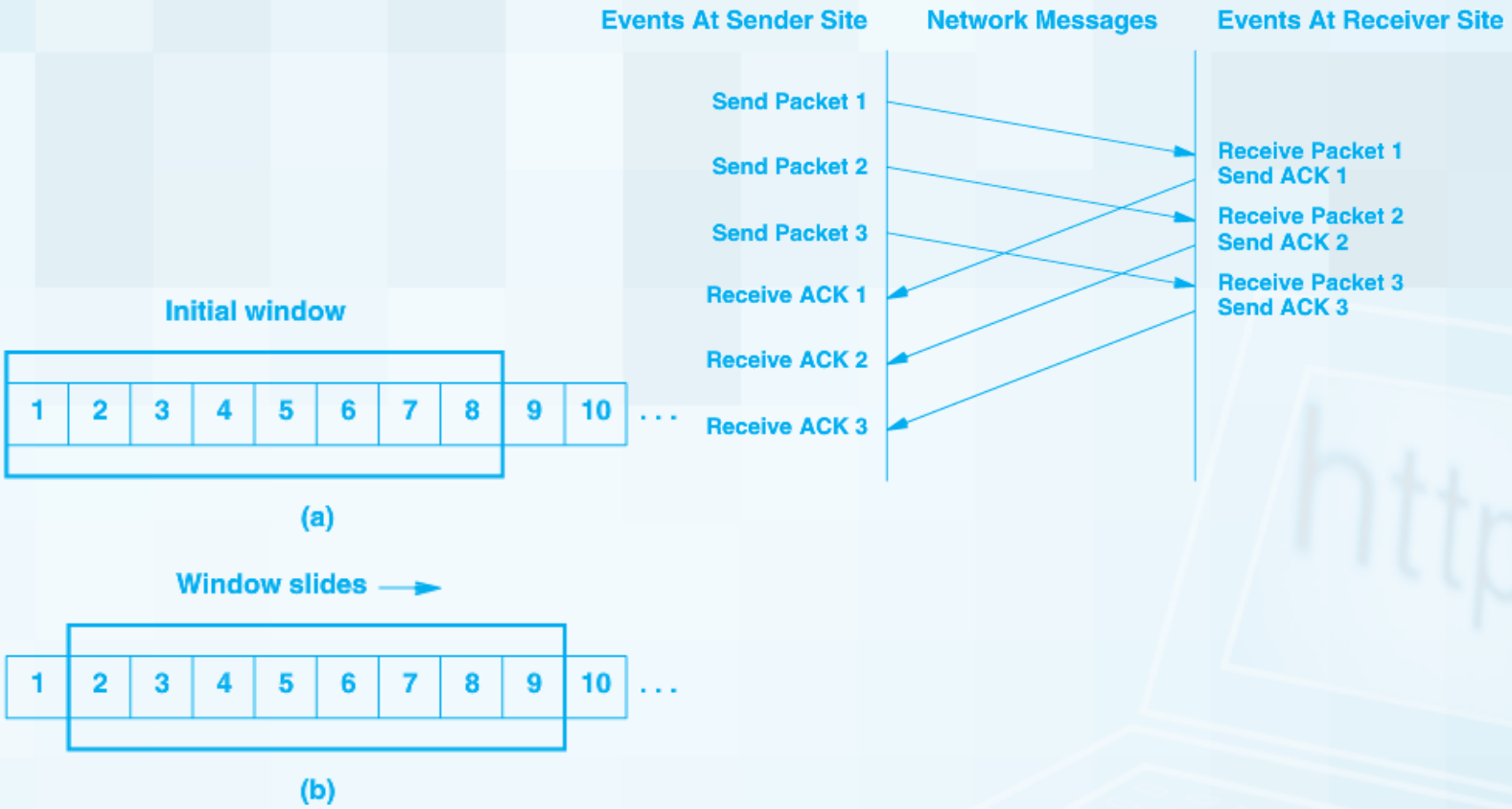


Packet Loss


Events At Sender Site
Network Messages
Events At Receiver Site
packet loss
Send Packet 1
Start Timer
Packet should have arrived
ACK would have been sent
ACK would normally
arrive at this time
Timer Expires
Retransmit Packet 1
Start Timer
Receive Packet 1
Send ACK 1
Receive ACK 1
Cancel Timer

Sliding Window - Pipelined

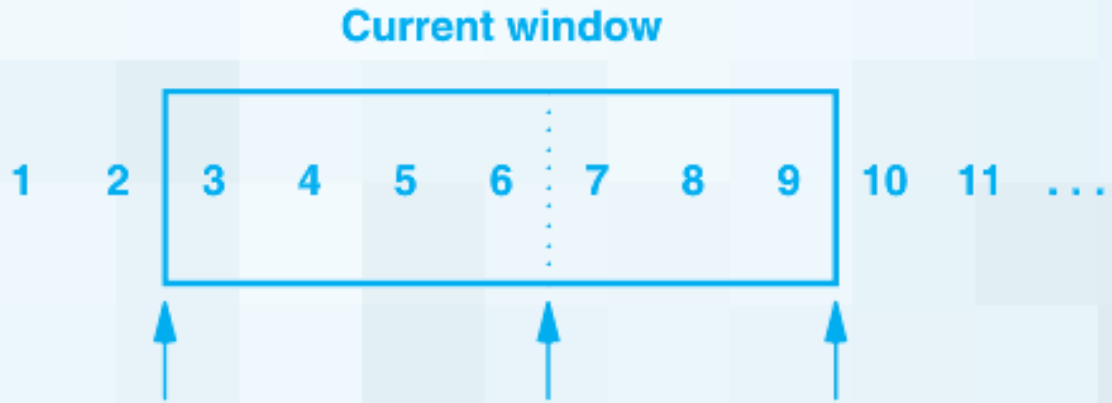
- Requires Buffering on each end



TCP Data Transfer Specifics

- Data transferred as a stream of octets
- Data is transferred in segments, but acknowledged at the octet level
- Full duplex – data can be transferred in either direction, or both
- Both endpoints of connection must maintain buffers/windows for both sending and receiving

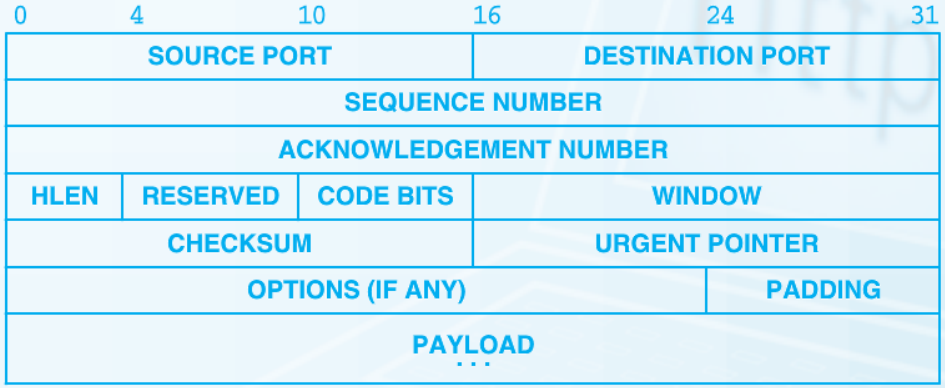
Sender's Window



- 1, 2 have been sent and acknowledged
- 3 – 6 sent but not acknowledged
- 7 – 9 have not been sent but can be without delay
- 10 and higher will not be sent until window moves

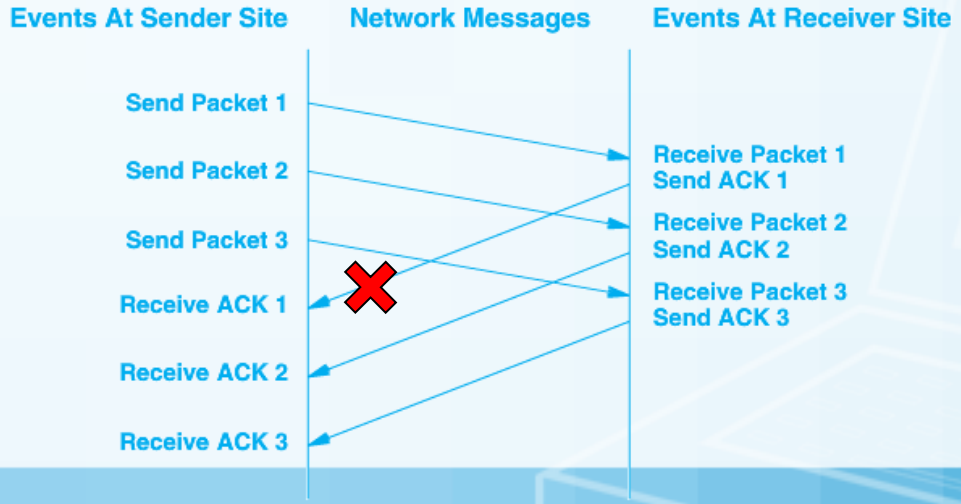
Window Advertisement

- Window size can vary over time
- Receiver sends a windows size with acknowledgement that indicates how many octets it is willing to accept
- Allows flow control
- An advertisement of 0 will halt transfer



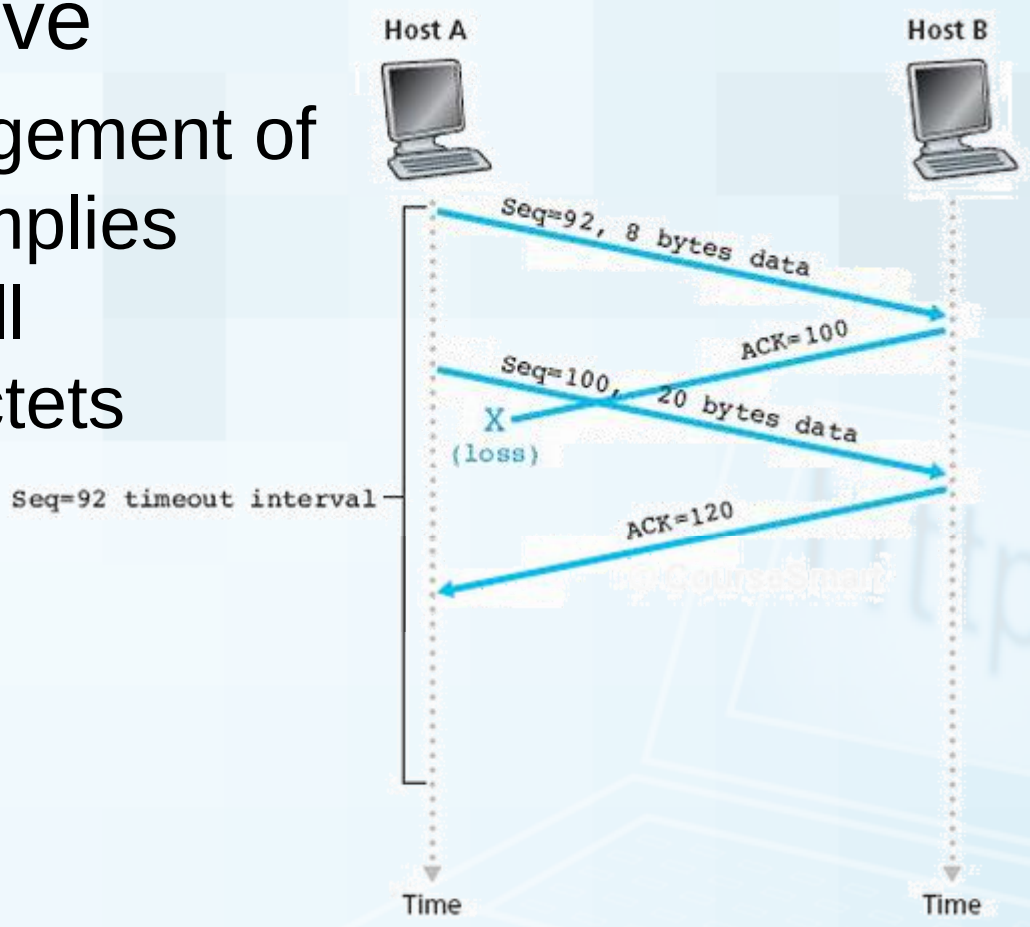
Acknowledgement

- Acknowledgements are cumulative
  - Acknowledgement of any octet implies receipt of all previous octets
  - ++ Simple
  - ++ Lost acknowledgements will not necessarily result in retransmission



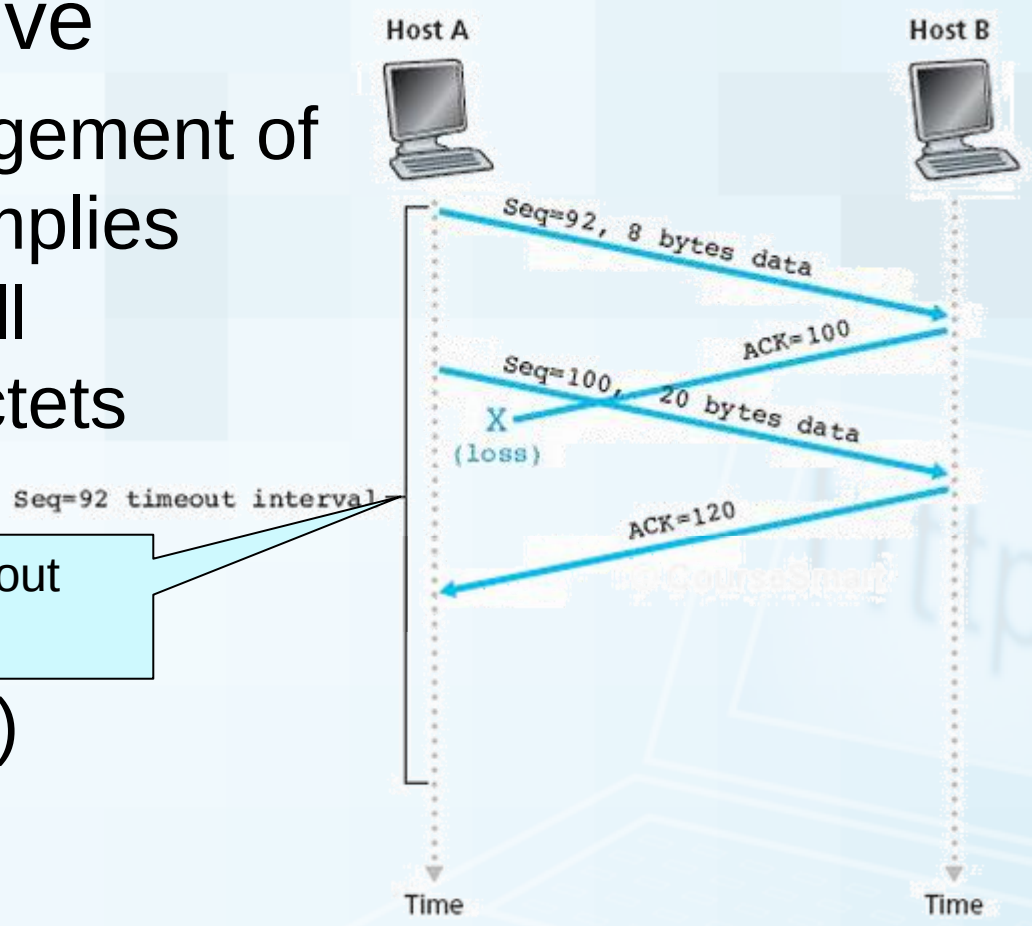
Acknowledgement

- Acknowledgements are cumulative
  - Acknowledgement of any octet implies receipt of all previous octets



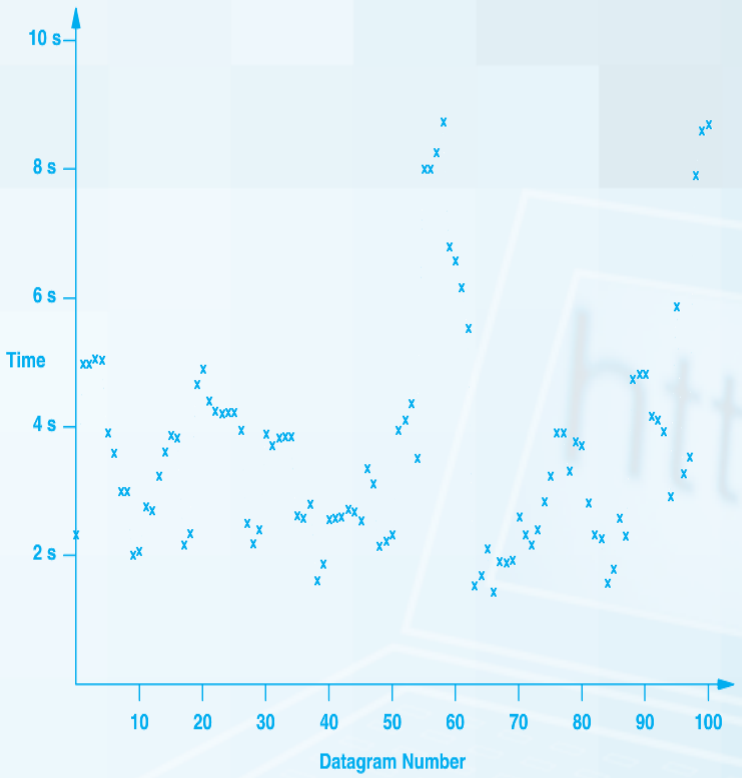
Acknowledgement

- Acknowledgements are cumulative
  - Acknowledgement of any octet implies receipt of all previous octets
  - Think (30 s)
  - Pair (30 s)
  - Shair



Timeout and Retransmission

- What do we use for a timeout?
  - LAN – round-trip time for ACK might be ms
  - Internet – 100x
  - Varies over time



Adaptive Retransmission

- Round-trip time (RTT) is monitored for each transmission/ACK

$$\text{EstimatedRTT} = (1 - \alpha) \cdot \text{EstimatedRTT} + \alpha \cdot \text{SampleRTT}$$

$0 \leq \alpha < 1$

Recommended value of $\alpha = 0.128$ [RFC 6298]

$$\text{DevRTT} = (1 - \beta) \cdot \text{DevRTT} + \beta \cdot |\text{SampleRTT} - \text{EstimatedRTT}|$$

Recommended β is 0.25

$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4 \cdot \text{DevRTT}$$

Congestion Control

- Flow control is a function of the receiver and its ability to accept data

- Congestion control is implemented by the sender to avoid excessive unsuccessful transmission (collapse)

New Variable - cwnd

- Congestion window

$$\underbrace{\text{LastByteSent} - \text{LastByteAcked}}_{\text{Un-acknowledged bytes}} \leq \min\{\text{cwnd}, \text{rwnd}\}$$

  - cwnd – congestion window
  - rwnd – receive window

We can send up to cwnd bytes per RTT period

cwnd

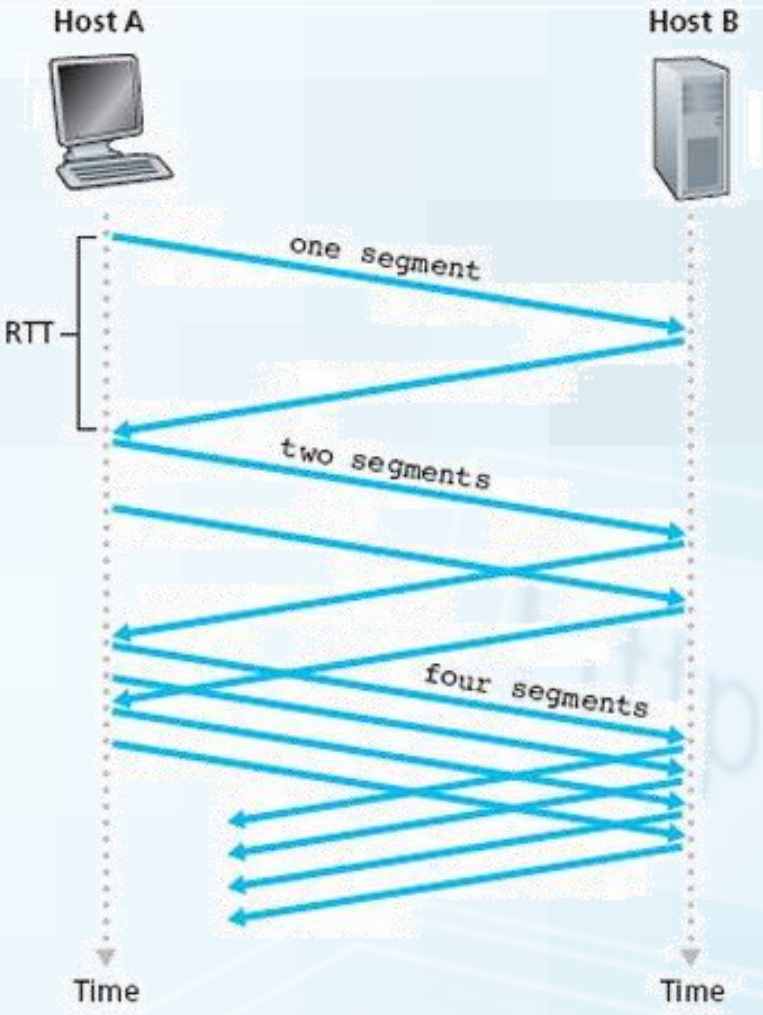
- Average transmission rate is roughly cwnd/RTT bytes/sec
- By manipulating cwnd, transmission rate can be controlled

Congestion Detection

- Essentially loss of segments
  - Retransmission on timeout
  - Fast retransmit on duplicate ACK
- Adjust cwnd
  - Decrease when a segment is lost
  - Increase when [consistent] ACKs are received
    - Continue to increase until a segment is lost, then backoff

TCP Slow Start

- Start with a small cwnd (one MSS)



cwnd Over Time

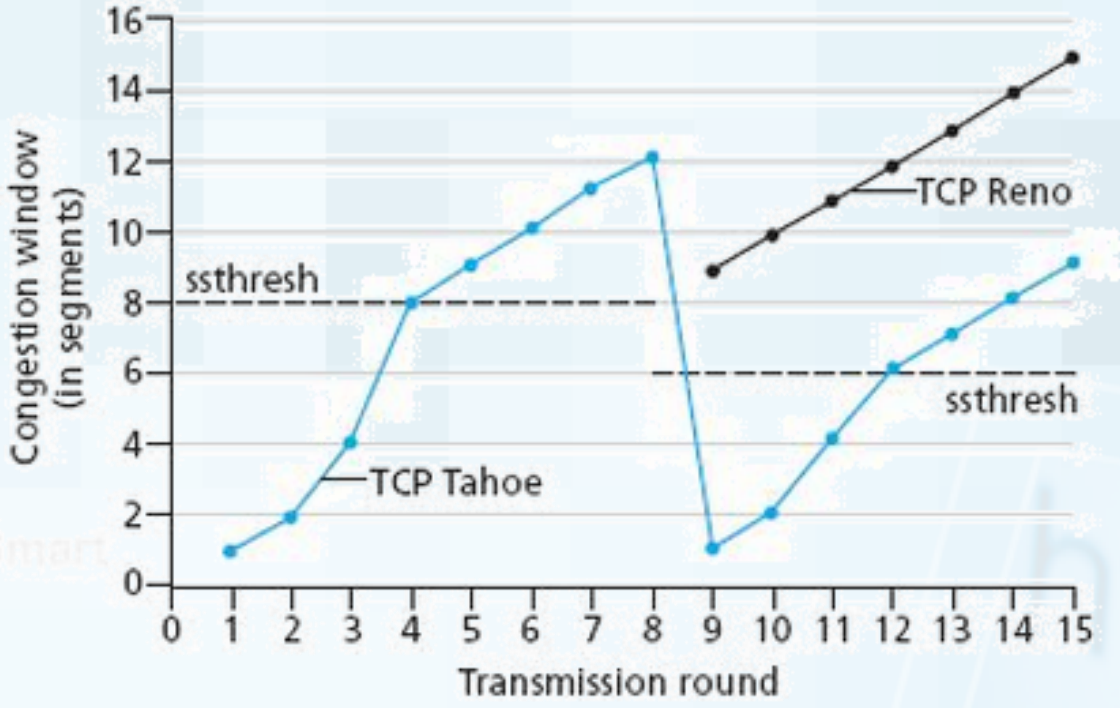


Figure 3.53 ♦ Evolution of TCP's congestion window (Tahoe and Reno)

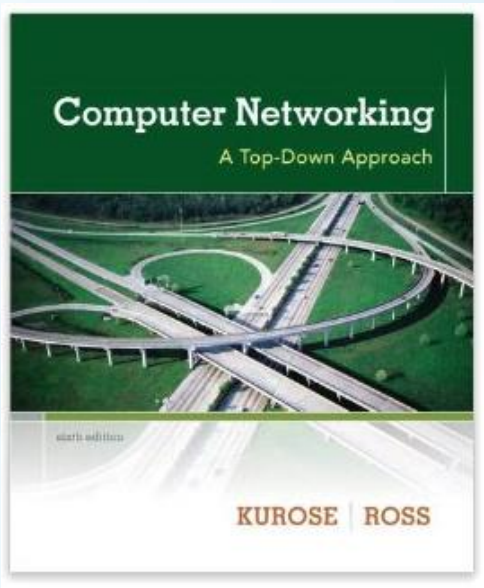
Lab Tomorrow

- Wireshark
- Your TCP Server / Client (from Lab #2)

The content of this video is based in part on lecture slides from a very good textbook, and used with the author's permission:

Computer Networking: A Top-Down Approach, 6e, by Jim Kurose and Keith Ross

Publisher: Pearson, 2013



It is also based on slides provided by Dr. Darrin Rothe